UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/886,455	06/20/2001	Jeffrey D. Washington	5150-48300	6993

35690 7590 07/14/2005

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.
P.O. BOX 398
AUSTIN, TX 78767-0398

EXAMINER	
VU; KIEU D	

ART UNIT	PAPER NUMBER
2173	

DATE MAILED: 07/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/886,455	WASHINGTON ET AL.	
	Examiner	Art Unit	
	Kieu D. Vu	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 April 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-7,10-23,26,29-38,40,43-46 and 49-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7,10-23,26,29-38,40,43-46 and 49-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-7, 10-23, 26, 29-38, 40, 43-46, and 49-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohara et al ("Ohara", USP 6366300) and Weeren et al ("Weeren", USP 5946485)

Regarding claims 1 and 34, Ohara teaches a computer-implemented method for programmatically generating a graphical program comprising displaying a graphical user interface (GUI) on a display (user interface on col 16, lines 20-29); receiving user input to the GUI specifying desired functionality (behavior characteristics) of the graphical program (selecting behavior) (line 66 of col 16 to line 10 of col 17); programmatically generating the graphical program in response to the user input specifying the functionality of the graphical program (automatic generation of a program) (col 16, lines 27-28), wherein the graphical program implements the specified functionality (col 5, lines 32-41). Ohara further teaches that the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17) and wherein said programmatically generating the graphical program includes

generating the block diagram portion (creating layout diagram) (col 7, lines 22-36, lines 55-59) (Fig. 37). Ohara neither teaches that the graphical program also comprises a graphical user interface portion nor teaches that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes. However, such features are known in the art as taught by Weeren. Weeren teaches that the graphical program also comprises a graphical user interface portion ("VerifyAcct" 402; Fig. 4) and that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes Fig. 6) (col 7, lines 20-30). It would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claims 2 and 35, Ohara teaches that the GUI comprises information useable in guiding a user in creation of a program (guidance display; col 44, lines 50-52)

Regarding claims 3 and 36, Ohara teaches the GUI comprises one or more GUI input panels and user input to each of the one or more GUI input panels (input panels of Fig. 4).

Regarding claims 4 and 37, Ohara teaches displaying a first GUI input panel on the display, wherein the first GUI input panel includes one or more first fields adapted to receive user input specifying first functionality of the graphical program (fields in Fig. 4);

receiving first user input specifying first functionality of the graphical program (specifying behavioral characteristics in Fig. 4); displaying a second GUI input panel on the display (Fig. 6), wherein the second GUI input panel includes one or more second fields adapted to receive user input specifying second functionality of the graphical program (fields in Fig. 6); receiving second user input specifying second functionality of the graphical program (setting output in Fig. 6).

Regarding claims 5 and 38, Ohara teaches the second GUI input panel is one of a plurality of possible second GUI input panels, wherein the second GUI input panel is displayed based on the first user input (click on "next operation" in Fig. 4 will lead to the window shown in Fig. 6) (col 17, lines 40-50).

Regarding claim 6, Ohara teaches generating a portion of a graphical program (Fig. 17).

Regarding claims 7 and 40, Ohara teaches generating the graphical program creates the graphical program without any user input specifying the new graphical program during said creating (default setting).

Regarding claims 10 and 43, Ohara teaches creating a plurality of graphical program objects in the graphical program; and interconnecting (wiring) the plurality of graphical program objects in the graphical program (col 23, lines 1-10) (also see Fig. 17) wherein the interconnected plurality of graphical program objects comprise at least a portion of the graphical program (see Fig. 17).

Regarding claim 11, Ohara teaches creating one or more user interface objects in the graphical program, wherein the one or more user interface objects perform one or more of displaying output from the graphical program (see Fig. 36) (col 44, lines 20-35)

Regarding claims 12 and 44, Ohara teaches the specifying an instrumentation function; wherein the programmatically generated graphical program implements the specified instrumentation function (col 65, lines 50-57).

Regarding claims 13 and 45, Ohara teaches an industrial automation function (col 65, lines 50-57).

Regarding claim 14, Ohara teaches calling an application programming interface (API) enabling the programmatic generation of a graphical program (col 16, lines 19-28).

Regarding claim 15, Ohara teaches requesting a server program (network connection) to generate the graphical program (col 25, lines 8-10)

Regarding claim 16, Ohara teaches a computer-implemented method for programmatically generating a graphical program, the method comprising displaying a plurality of GUI input panels on a display (col 16, lines 20-29) (Fig. 4 and Fig. 6), wherein the GUI input panels comprise information useable in guiding a user in creation of a program (guidance display); receiving user input to the plurality of GUI input panels, wherein the user input specifies desired functionality of the graphical program (selecting behavior) (line 66 of col 16 to line 10 of col 17); programmatically generating the graphical program in response to the user input specifying the functionality (automatic generation of the program) (col 16, lines 27-28) of the graphical program, wherein the

graphical program implements the specified functionality (col 5, lines 32-41). Ohara further teaches that the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17) and wherein said programmatically generating the graphical program includes generating the block diagram portion (creating layout diagram) (col 7, lines 22-36, lines 55-59) (Fig. 37).

Ohara neither teaches that the graphical program also comprises a graphical user interface portion nor teaches that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes. However, such features are known in the art as taught by Weeren. Weeren teaches that the graphical program also comprises a graphical user interface portion ("VerifyAcct" 402; Fig. 4) and that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes Fig. 6) (col 7, lines 20-30). It would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claim 17, Ohara teaches a computer-implemented method for programmatically generating a graphical program comprising displaying a graphical user interface (GUI) on a display (user interface on col 16, lines 20-29); receiving user input to the GUI specifying desired functionality of the graphical program (behavior characteristics); executing a graphical program generation (GPG) program (automatic

generation of a program) (col 16, lines 27-28); the GPG program receiving the user input, wherein the user input specifies the desired functionality of the new graphical program (selecting an output signal behavior) (col 16, lines 29-40); the GPG program programmatically generating the graphical program in response to the user input specifying the functionality of the graphical program, wherein the graphical program implements the specified functionality (col 5, lines 32-41). Ohara further teaches that the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17) and wherein said programmatically generating the graphical program includes generating the block diagram portion (creating layout diagram) (col 7, lines 22-36, lines 55-59) (Fig. 37).

Ohara neither teaches that the graphical program also comprises a graphical user interface portion nor teaches that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes. However, such features are known in the art as taught by Weeren. Weeren teaches that the graphical program also comprises a graphical user interface portion ("VerifyAcct" 402; Fig. 4) and that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes Fig. 6) (col 7, lines 20-30). It would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claim 18, Ohara teaches a computer-implemented method for programmatically generating a graphical program comprising displaying a graphical user interface (GUI) on a display (user interface on col 16, lines 20-29); receiving user input to the GUI specifying desired operation of the graphical program (behavior); executing a graphical program generation (GPG) program (automatic generation of a program) (col 16, lines 27-28); the GPG program receiving the user input, wherein the user input specifies the desired operation of the new graphical program (selecting an output signal behavior) (col 16, lines 29-40); the GPG program programmatically generating the graphical program in response to the user input specifying the operation of the graphical program, wherein the graphical program implements the specified operation (col 5, lines 32-41). Ohara further teaches that the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17) and wherein said programmatically generating the graphical program includes generating the block diagram portion (creating layout diagram) (col 7, lines 22-36, lines 55-59) (Fig. 37).

Ohara neither teaches that the graphical program also comprises a graphical user interface portion nor teaches that the program is generated automatically without direct user input specifying the plurality of nodes or connections between the nodes. However, such features are known in the art as taught by Weeren. Weeren teaches that the graphical program also comprises a graphical user interface portion ("VerifyAcct" 402; Fig. 4) and that the program is generated automatically without direct user input

specifying the plurality of nodes or connections between the nodes Fig. 6) (col 7, lines 20-30). It would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claim 19, Ohara teaches the GPG program comprises a graphical programming development environment application (col 1, lines 19-28).

Regarding claim 20, Ohara teaches the generating a plurality of graphical programs, depending on the received user input (inherent).

Regarding claim 21, Ohara teaches a method for programmatically generating a graphical program, the method comprising displaying one or more input panels on a display (Fig. 4); receiving user input to the one, or more input panels (inputs to window in Fig. 4); programmatically generating graphical source code for the graphical program, based on the received user input (col 25, lines 5-8). Ohara further teaches that the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17). Ohara does not teach automatically generating the graphical source code for the block diagram portion without direct user input specifying the plurality of nodes or connections between the nodes. However, such features are known in the art as taught by Weeren. Weeren teaches automatically generating the graphical source code for the block diagram portion without direct user input specifying the plurality of nodes or connections between the nodes (Fig. 6) (col 7, lines 20-30). It

would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claim 22, Ohara teaches input panels comprise a graphical user interface (GUI) (user interface on col 16, lines 20-29) useable in guiding a user in specifying program functionality (guidance display; col 44, lines 50-52); wherein the received user input specifies desired functionality of the graphical program (selecting behavior) (line 66 of col 16 to line 10 of col 7) wherein the programmatically generated graphical source code implements the specified desired functionality (col 25, lines 5-8).

Regarding claim 23, Ohara teaches a computer-implemented method for programmatically generating a graphical program comprising displaying a node (graphical object) in the graphical program in response to user input (col 16, lines 41-47); displaying a graphical user interface (GUI) for configuring functionality for the node in response to user input and receiving user input via the GUI indicating desired functionality for the node (line 66 of col 16 to line 6 of col 17); programmatically including graphical source code associated with the node in the graphical program, wherein the programmatically included graphical source code implements the desired functionality (col 25, lines 5-8) wherein the graphical source code comprises a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17). Ohara does not teach automatically generating the graphical source code as a sub-program of the graphical program without direct user

input specifying the plurality of nodes or connections between the nodes, wherein the node represents the sub-program. However, such features are known in the art as taught by Weeren. Weeren teaches automatically generating the graphical source code as a sub-program of the graphical program without direct user input specifying the plurality of nodes or connections between the nodes (Fig. 6) (col 7, lines 20-30), wherein the node represents the sub-program (icons in Fig. 4 represent sub-programs). It would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claims 26 and 46, Ohara teaches steps for configuring a node in a graphical program comprising displaying a node (graphical object) in the graphical program in response to user input (col 16, lines 41-47); displaying a graphical user interface (GUI) for configuring functionality for the node in response to user input and receiving user input via the GUI indicating desired functionality for the node (line 66 of col 16 to line 6 of col 17); programmatically including graphical source code associated with the node in the graphical program, wherein the programmatically included graphical source code implements the desired functionality (col 25, lines 5-8) wherein the graphical source code comprises a plurality of interconnected nodes (graphical objects) that visually indicate functionality of the graphical program (see Fig. 17)

Ohara does not teach automatically generating the graphical source code as a sub-program of the graphical program without direct user input specifying the plurality of

nodes or connections between the nodes, wherein the node represents the sub-program. However, such features are known in the art as taught by Weeren. Weeren teaches automatically generating the graphical source code as a sub-program of the graphical program without direct user input specifying the plurality of nodes or connections between the nodes (Fig. 6) (col 7, lines 20-30), wherein the node represents the sub-program (icons in Fig. 4 represent sub-programs). It would have been obvious to one of ordinary skill in the art to apply Weeren teaching in Ohara graphical program environment so that a program can be automatically generated without user's manual connections between nodes (Weeren, col 2, lines 17-21).

Regarding claims 29, 31-32, and 49, it is inherent that Ohara teaches no graphical source code or no functionality or no program instructions is associated with the node until after said programmatically generating graphical source code associated with the node.

Regarding claims 30 and 50, Ohara teaches default graphical source code is associated with the node (52, lines 3-7).

Regarding claims 33 and 51, Ohara teaches the changing functionality of the node (col 5, lines 47-53), after said programmatically generating the graphical source code; re-displaying the GUI in response to the user input requesting to change functionality of the node (col 16, lines 20-29); receiving user input to the GUI specifying new functionality for the node (inputs to window in Fig. 4); programmatically replacing

the previously generated graphical source code with new graphical source code to implement the new functionality for the node (col 25, lines 5-8).

Regarding claim 52-53, Weeren teaches that the graphical user interface portion comprises an indicator (see "VerifyAcct" in Fig. 4)

3. Applicant's arguments filed on 04/11/05 have been fully considered but they are moot under new ground of rejection.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kieu D. Vu.

The examiner can normally be reached on Mon - Thu from 7:00AM to 3:00PM at 571-272-4057.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca, can be reached at 571-272-4048.

The fax phone numbers for the organization where this application or proceeding is assigned are as follows:

571-273-8300

and / or:

571-273-4057 (use this FAX #, only after approval by Examiner, for "INFORMAL" or "DRAFT" communication. Examiners may request that a formal paper / amendment be faxed directly to them on occasions).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703-305-3900).

Kieu D. Vu